An Analytic Comparison of α-False Eye Separation, Image Scaling and Image Shifting in Stereoscopic Displays

Zachary Wartell, Larry F. Hodges, William Ribarsky

An Analytic Comparison of α-False Eye Separation, Image Scaling and Image Shifting in Stereoscopic Displays

Zachary Wartell, Larry F. Hodges, William Ribarsky

Abstract

Stereoscopic display is a fundamental part of many virtual reality systems. Stereoscopic displays render two perspective views of a scene, each of which is seen by one eye of the user. Ideally the user's natural visual system combines the stereo image pairs and the user perceives a single 3D image. In practice, however, users can have difficulty fusing the stereo image pairs into a single 3D image. Researchers have used a number of software methods to reduce fusion problems. Some fusion algorithms act directly on the 3D geometry while others act indirectly on the projected 2D images or the view parameters. Compared to the direct techniques, the indirect techniques tend to alter the projected 2D images to a lesser degree. However while the 3D image effects of the direct techniques are algorithmically specified, the 3D effects of the indirect techniques require further analysis. This is important because fusion techniques were developed in non-head-tracked displays that have distortion properties not found in the modern head-tracked variety. In non-head-tracked displays, the non-head-tracked distortions can mask the stereoscopic image artifacts induced by fusion techniques but in head-tracked displays distracting effects of a fusion technique may become apparent. This paper is concerned with stereoscopic displays in which the head is tracked and the display is stationary, attached to a desk, tabletop or wall. This paper rigorously and analytically compares the distortion artifacts of three indirect fusion techniques, α-false eye separation, image scaling and image shifting. We

show that the latter two methods have additional artifacts not found in α-false eye separation and we conclude that α-false eye separation is the best indirect method for these displays.

1 Introduction

Virtual environments aim to perceptually place the user in a computer-generated world. A key component of creating this illusion is interactive 3D imagery. To generate this imagery, a typical VR system has a location and orientation tracking device, an image generator and one or more displays. The tracking device determines the positions of the user's head and/or eyes and of the displays. The image generator computes the image that each eye would see on a display surface if the eye and the display existed inside the virtual world at their tracked positions. This image is then fed to the display. VR systems are typically configured either as a head-mounted display (HMD) or as a head-tracked display (HTD). In a HMD, the display is attached to a helmet or headset worn by the user, so both the eye points and the display are in continuous motion. In a HTD, the display is stationary, attached to a desk, tabletop, or wall. Hence only the eye points move. HTD examples are the CAVE [1], desktop VR [27], and the responsive workbench [13].

Most VR systems generate a pair of images, one for each eye. This stereoscopic imagery provides a true 3D image so virtual objects appear to exist in front of and behind the physical display surface. Software methods for stereoscopic display are well known[8][23][18]. Stereoscopic display for virtual reality has been shown to improve user depth perception and task performance in a variety of tasks [20][27][3]. This is not surprising since real world experience shows that stereopsis is an important depth cue, especially for objects within the user's personal space (1.5 meters) [2].

Stereoscopic HTDs add additional challenges to interface design. A key issue is maintaining good stereoscopic viewing conditions. This requires balancing several goals [33] including the fusibility of the stereoscopic images Both experience [14] and experimental studies [39][24] have shown that users with normal stereoscopic vision often have trouble fusing stereo image pairs into a single 3D image even when the viewing geometry is modeled exactly. Users may experience headaches, eyestrain and/or fatigue. At extremes they may be unable to fuse the images into a single 3D perception.

To maintain good stereoscopic viewing conditions, the system must dynamically adjust the user's view of the environment as she travels through and manipulates the virtual world. Numerous degrees of freedom control the view. We partition these into: view placement, view scale and view optics. View placement refers to the location and orientation of the projection window. The projection window is the virtual representation of the HTD's physical display surface in the virtual world. View placement does not refer to eye point locations because in a HTD the user's head position is a physical parameter controlled by the user and is not under software control. View scale is a single degree of freedom that represents the viewer's size in the world. Collectively, we refer to view placement and scale as the extrinsic parameters. View optics includes all other parameters modeled by the pin-hole camera model in interactive computer graphics This includes modeled eye separation, the position of the near and far clipping planes, field of view, and other distortions such as depth compression or expansion.

For controlling image fusion problems in non-head-tracked displays prior researchers used the following view optics techniques:

(1) 'underestimated eye separation' – This method sets the modeled eye separation to underestimated value either statically [8] or dynamically [28][29].

(2) ‘image scaling’ –This method scales down the projected 2D images about the center of the screen (called frame magnification in [22]).

(3) ‘image shifting’ – This methods translates the left and right eye projected 2D images towards each other. Its use to alter the stereo image dates back to early stereo photography of the 1800’s [26, p201-202]. Hodges and McAllister [10] discuss it with respect to computer graphics and Siegel, Tobinaga, and Akiya [21] use it to manage image fusion in non-head-tracked, stereoscopic video.

(4) ’fusibility clipping’ – This method sets the near and far clipping planes so as to clip out unfusible geometry [23].

(5) ‘optic scaling’ – This method scales the world perpendicular to the projection plane to bring objects closer to the projection plane prior to 2D projection [36].

(6) ‘asymmetric/asymptotic technique’ – Williams and Parrish [36] contrast optic scaling with prior techniques which map the z-coordinate asymptotically. They give an example ‘asymmetric/asymptotic’ matrix with this behavior. However, the matrix also embeds the standard local-to-screen space transform and the matrix is not general enough to be used for a head at arbitrary position and orientation as is necessary in a head-tracked display.

Additionally researchers have used extrinsic adjustment techniques, mainly view scaling. This method changes the viewer’s size to *uniformly* scale the perceived world so that geometry is at fusible distances. For non-head-tracked displays, Ware et. all [28][29][31] mix viewer scaling with underestimated eye separation. For certain applications, view scaling and view placement is sufficient [30][33].

Understanding the distortion effects of optic scaling and viewer scaling is straightforward because these algorithms explicitly specify a 3D transformation. However, understanding the distortion effects of the other methods is non-trivial since they work indirectly on the 2D projected images or on the view optic parameters. Empirically, the indirect methods tend to have much more subtle and less noticeable effects on the monoscopic image properties which can be seen when viewing only a single eye's image. False eye separation, for example, has negligible monoscopic effects since it involves only an inward movement of the eye points. Image shifting has no effects on the monoscopic image properties. Finally while 'fusibility clipping' does not distort the world at all, it is a culling method that throws out geometry. All other methods are non-culling methods. Clearly not all applications can tolerate this throwing away of geometric information, so non-culling methods are a vital tool.

While any view optic technique to reduce fusion problems will distort the perceived image, the question is what are the artifacts, or the distracting and undesirable aspects, of a technique's distortion, and how do these artifacts compare to those of other techniques. Some artifacts maybe static and noticeable with no head motion while other artifacts maybe dynamic and only noticeable with head motion. Also note that in a stereoscopic HTD the user is free to view the displays from an arbitrary angle. While often a user's eye axis tends to remain close to parallel to the screen in a single display system, this is not guaranteed. Moreover, in many multi-display systems the eye axis will always be non-parallel to some display surface. For these reasons, a general purpose method for fusion control should have minimal artifacts in the non-parallel case too.

In [32] we show that false eye separation modeling distorts the perceived 3D image by a non-affine homology (a collineation that does not preserve parallelism). Moreover, this homology

has a head-position dependent shearing component that causes the perceived 3D image to shift side to side with left/right head movement. Additionally, head movement perpendicular to the display results in compress and expansion of the 3D image. These effects occur despite *perfect* head-tracking. In [34], we derive a method to remove the side to side shifting inherent in the false eye modeling techniques. In this paper we refer to this augmented technique as α-false eye separation.

This paper will show that in the parallel case, image scaling has dynamic artifacts similar α-false eye separation plus an additional static artifact. In the non-parallel case, image scaling has even more artifacts not found in α-false eye separation nor standard false eye separation. This paper will also show that in the parallel case image shifting has additional dynamics artifacts beyond α-false eye separation and that in the non-parallel case image shifting has even further artifacts. We conclude that α-false eye separation is the best of the three indirect techniques for stereoscopic HTDs.

2 Background and Previous Work

When a user cannot fuse a stereo image pair into single 3D image, she experiences diplopia (double vision). In a stereoscopic display the occurrence of diplopia is related to various physical attributes of the display system and the geometry of the display environment[8]. The relevant geometric aspects are:

- the distance of the displayed virtual object relative to the display surface
- the eye separation value used in computing the viewing transform
- the distance of the user's eyes to the display surface

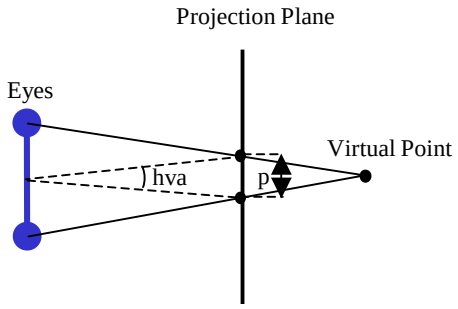


Figure 1: Illustration of the projection of a virtual point onto the projection plane for the two eyes of a user. *P* is the screen parallax, or distance on the screen between the stereo images of a virtual point. *Hva* is the horizontal visual angle, the visual angle subtended by *p*.

Figure 1 illustrates two important measurements. The eyes are on the left and a point on a virtual object is on the right. This point is projected onto two points on the projection plane in the middle. The two projected points which correspond to a single 3D point are called homologous points. The screen parallax, *p*, associated with a virtual 3D point is the distance between these projected 2D image points. Related to the screen parallax is the horizontal visual angle or HVA. HVA is the visual angle subtended by the screen parallax. For virtual points in front of the screen, screen parallax and HVA are defined to be negative while for virtual points behind the screen, screen parallax and HVA are defined to be positive. The negative values are called "crossed" values since eyes which are fixated on the screen must cross to fixate on a point in front of the screen. The positive values are called "uncrossed" values since eyes which are fixated on the screen must uncross to fixate on a point behind the screen.

Research has shown that if the HVA of homologous points is outside a certain range then diplopia occurs and the 3D depth illusion collapses [39]. Yeh and Silverstein found an average

range of [-4.93,1.57] for a stereo CRT. Even if diplopia does not occur, additional problems such as fatigue, eyestrain, and headaches may occur [14]. Mon-Williams, Wann, and Rushton [16] show that using a stereoscopic display can temporarily alter the visual system's internal coupling of accommodation (eye focus) and convergence (the relative orientation of one eye to the other). Collectively, we refer to these problems as 'image fusion problems.'

The indirect fusion control techniques, underestimated eye separation, image shifting and image scaling, reduce image fusion problems by bringing homologous points closer together. This reduces the screen parallaxes of the stereo image pair and reduces image fusion problems. These methods were undoubtedly experimented with by stereoscopic photographer's as far back as the 1800's [26] [15]. In the context of computer graphics, these methods represent additional parameters in the viewing geometry. Additionally software can dynamically vary and optimize a given method's parameters based on the virtual scene content [28][29][31]. With modern image processing, it is even possible to apply such a dynamic implementation to stereoscopic video recordings of the real world [12].

In [32] we observed and analyzed the effects of false eye separation modeling (either underestimated or overestimated) in a stereo HTD. [32] derives and discusses an analytic description of this distortion. The distortion is a non-affine collineation so while lines are mapped to lines, neither distances, angles nor parallelism is preserved. Moreover, as the user moves her head laterally (i.e. parallel to the projection plane) the perceived 3D image will shear laterally. As the user moves her head perpendicular to the screen (i.e. towards or away from the screen), the perceived 3D image will compress and expand. The latter two effects are most disappointing for a stereo HTD. One of the reasons for adding head-tracking to a stereoscopic display is remove exactly these types of distortions which had been observed in earlier non-head-

tracked systems. A user viewing a static, non-tracked stereo image that is rendered using correct eye separation will perceive the 3D image to warp and shear as she moves her head [25][8]. Head-tracking with correct eye separation removes this warping and yields a rigid 3D image under user head movement. Introducing false eye separation to a stereo HTD introduces induced stereomovement all over again!

In [34] we decompose the matrix representing this distortion and extract and invert the shear component. We then dynamically place the inverted shear on the matrix stack at run-time. This inverted shear predistortion cancels out the shear component of the basic distortion. The result is that the perceived 3D image no longer shifts side to side with lateral head movement. The expand/compression due to forward/backward head movement remains however. A slight modification of the shear predistortion matrix yields a more flexible predistortion matrix denoted α which adapts the end result to different physical display configurations. This augmented false separation method is called α-false eye separation.

Image scaling scales down the projected points about the center of the screen and image shifting translates the left and right images toward one and another. If done carefully this reduces the overall maximum screen parallax in the screen. Both methods were developed on non-head-tracked systems. While we are aware of published literature addressing issues in stereoscopic display distortions [8][4][17][9][38][28][32][26], none address the 3D effects on the 3D perceived image due to image scaling or image shifting with respect to stereo HTDs. It has been qualitatively noted, however, that image shifting shifts the perceived 3D depth of virtual objects [26][5][11] and Hodges [10] quantitatively describes how image shifting translates the stereoscopic window.

3 Methodology

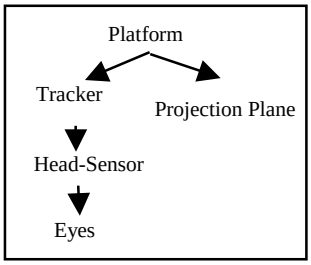


Figure 2: The coordinate system hierarchy for a typical head-tracked display.

To derive an analytic description of stereoscopic distortions, we must concisely describe the viewing model used in stereoscopic HTD's. A typical viewing model consists of the coordinate system hierarchy presented in Figure 2. The top coordinate system is the Platform Coordinate System. Manipulating this coordinate system moves the user through the virtual space. Directly attached to this coordinate system is the Projection Plane Coordinate System and the Tracker Coordinate System. The Projection Plane Coordinate System contains the projection plane in its XY plane with the projection window centered about the origin. The Tracker Coordinate System simply represents the tracker's emitter. Attached to the Tracker Coordinate System is the Head-Sensor Coordinate System and attached to that is the Eyes Coordinate System. The two eye points are on the x-axis of the Eyes Coordinate System and are symmetric about the origin.

The position and orientation of each child coordinate system relative to its parent are measured physically from the physical display setup as are the view window dimensions. The platform coordinate system's mapping to virtual world coordinates defines the mapping of the physical space of the real world to the virtual space of the virtual world. In addition to specifying the position and orientation, the Platform Coordinate System can also be uniformly scaled. This causes the virtual world to appear to grow and shrink.

This scale factor requires that we distinguish between physical and virtual distance measurements. We assume that all measurements such as eye separation and window size are made in physical units. Each of these distances has a corresponding value in virtual space which is computed by simply multiplying the physical value by the platform scale factor. For example if the eye separation is measured as 0.06 m and the platform scale is 1000, then the virtual eye separation is 60 m. In this paper, whenever, we discuss distance measurements such as eye separation, we are always referring to the physical values not the virtual ones. Importantly, the term false eye separation refers to a discrepancy between the true physical eye separation and the modeled physical eye separation and *not* the discrepancy between physical and virtual measurements.

When analyzing stereoscopic distortions we assume that all the important physical measurements are correct unless otherwise noted. This includes those made dynamically by the tracking system. We also assume any distortion due to curvature of the screen or any optics is negligible or accounted for by other means [4]. Additionally, we assume that the natural change of the eye separation as measured between the entrance pupils of the human eyes [19] is also relatively negligible.

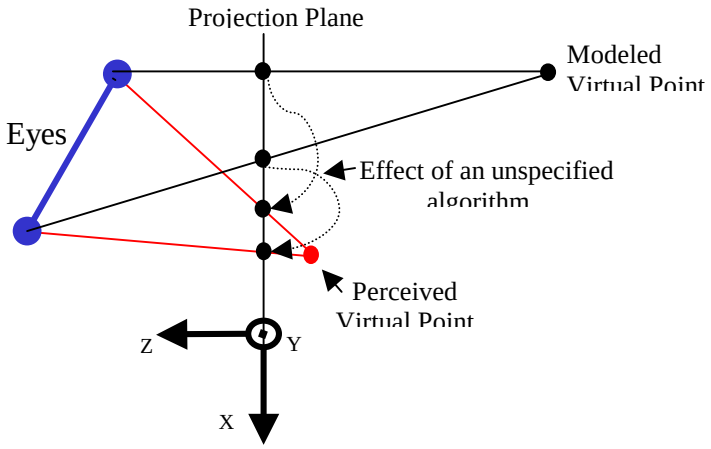


Figure 3: This diagram is a geometric construction describing the distortion of perceived space due to an generic image fusion algorithm. The diagram is an overhead view of a user viewing a stereo HTD. The eye points are in blue. The projection plane is the vertical black line. A modeled virtual point is projected onto the projection plane via the black projectors, but the image fusion algorithm displays transformed versions of these projected points. The user perceives a 3D point at a different location, the perceived virtual point.

These assumptions allow us to independently analyze the distortions in which we are interested. We can describe stereoscopic distortions using simple geometric constructions such as Figure 3. Figure 3 is a highly abstract diagram of a user viewing a stereoscopic display. The diagram is drawn from an overhead point of view looking down on the user. The user is represented by her eye points in blue and is looking at a display screen which is drawn as the vertical black line labeled 'Projection Plane.' The illustrated coordinate system is the Projection Plane Coordinate System. The projection plane is in the XY plane and the projection window is centered about the origin. (Note, Figure 3 only shows a portion of the projection window so the window does not actually appear centered in the diagram). A single modeled point on a virtual object is shown

projected onto the projection plane once for each eye via its projectors in black. The image fusion techniques we are concerned with distort the projected image points to new points in the projection plane. This is indicated by the dotted arrows between the original projected image points and the altered image points. These new points are displayed to the user. As a result the user perceives the 3D point at a different location, called the perceived point (red). By applying the construction to every point in the modeled scene we can graphically depict the perceived scene.

4 Geometric Construction

In this section we develop specific geometric constructions describing the distortion due to image scaling and image shifting. These will lead to the analytic descriptions in Section 5.

4.1 Image scaling

The suggested implementation for image scaling (called frame magnification in [22]) is to compose the following matrices:

$$M = M_{scr}\, S_{mag}\, M_{proj}\, M_{view}\, M_{model} \qquad (1)$$

In our notation, $\boldsymbol{M_{model}}$ maps model coordinates to world coordinates. $\boldsymbol{M_{view}}$ maps world coordinates to view coordinates. $\boldsymbol{M_{proj}}$ maps view coordinates to the canonical projection coordinates and $\boldsymbol{S_{mag}}$ is the image scaling. $\boldsymbol{M_{scr}}$ maps canonical projection coordinates to the device dependent screen coordinates. (Note, Southard's notation uses row vector notation so our presentation is the reverse order of his, and also he combines the image scaling scale, $\boldsymbol{S_{mag}}$, and $\boldsymbol{M_{proj}}$ into a single matrix which he labels $\boldsymbol{N_{proj}}$.)

$\boldsymbol{M_{scr}}$ contains a scales and translations [6, pg278] and is invertable. So clearly:

$$M_{scr}\, S_{mag} = (M_{scr}\, S_{mag}\, M_{scr}^{-1})\, M_{scr} \qquad (2)$$

The latter equation is simply a scale about the center of the final window in screen coordinates. Assuming that all components of the viewing hierarchy are correctly measured, a scale about the screen window center is equivalent to a scale about the Projection Plane Coordinate System origin (Figure 4).

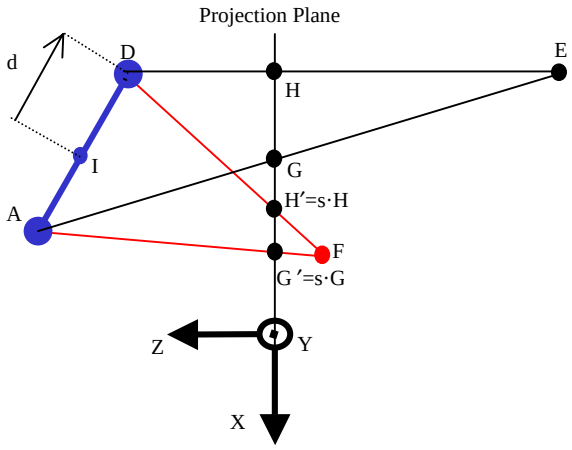


Figure 4: This is an abstract diagram of a user viewing a stereoscopic HTD. While this diagram is more detailed it uses the same color conventions established in Figure 3. **I** is the center of the eye axis. **D** is the left eye, displaced by **d** from **I**. **A** is the right eye displaced by -**d** from **I**. The projection plane is embedded in the X-Y plane of the illustrated Projection Plane Coordinate System. **E** is a point of modeled geometry. It is projected onto points **H** and **G**. Image scaling scales **H** and **G** to points *s*•**H** and *s*•**G**. The user's natural visual system reconstructs a perceived 3D point at **F**.

We can describe the stereoscopic distortion induced by image scaling with the geometric construction illustrated in Figure 4. Figure 4 is an abstract diagram of a user viewing a stereoscopic display. The diagram is drawn from an overhead point of view looking down on

the user using the same color conventions established in Figure 3. (Note, Figure 4 only shows a portion of the projection window so the window does not actually appear centered in the diagram). **I** is the eye axis center. **D** is the left eye displaced by vector **d**. **A** is the right eye displaced by vector –**d**. **E** is a modeled point on a virtual object. The modeled point is first projected on the projection window to points **H** and **G**. Image scaling then scales these points by factor *s* about the origin of the coordinate system. This yields points $\mathbf{H}'=s\cdot\mathbf{H}$ and $\mathbf{G}'=s\cdot\mathbf{G}$. These scaled points are the points actually displayed to the user. Ideally, the user's natural visual system then reconstructs the perceived 3D point at location **F**. Unfortunately image scaling introduces a problem which is not evident in this 2D diagram: the red rays $\mathbf{AG}'$ and $\mathbf{DH}'$ do not generally intersect when the eye axis is not parallel to the projection plane. To deal with this complication, we analyze the parallel case and the more general, non-parallel case separately.

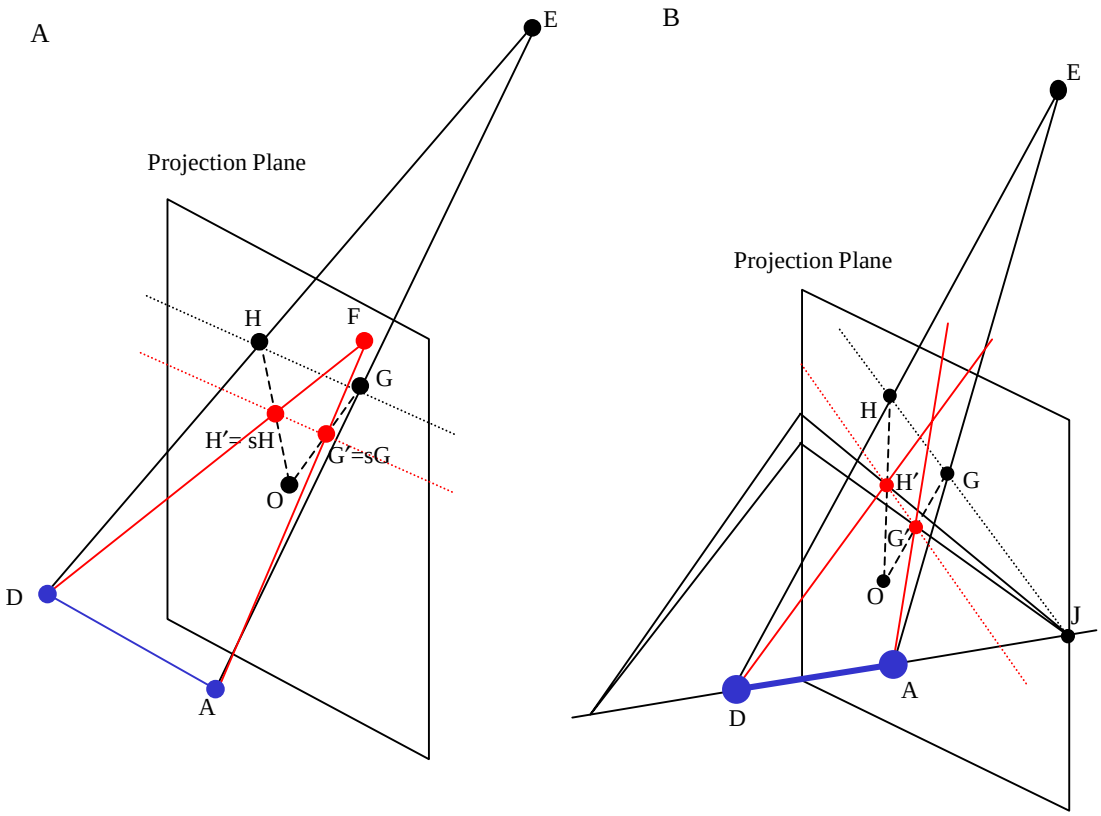


Figure 5: These figures illustrate a modeled point **E** being projected onto the projection plane once for each eye. The eyes are in blue and the projected points are **H** and **G**. These points are scaled about the center of the projection plane, **O**. The new points *s***H** and *s***G** are the points that the user sees. If the eye axis is parallel to the projection plane as in Figure A the reconstructed point **F** is well-defined. If the eye axis is not parallel to the projection plane as in Figure B, the reconstructing rays (red) may not intersect which leaves **F** undefined.

First we verify that in the parallel case the rays **AG′** and **DH′** do intersect. Figure 5A illustrates the parallel case construction in 3D. **O** is the origin of the Projection Plane Coordinate System. **E** projects onto **H** and **G**. Since **AD** is parallel to the plane, **HG** is parallel to **AD**. (This occurs since for a line *l* (here **AD**) parallel to a plane *p* (here the projection plane),

any plane *q* (here **ADGH**) containing *l* intersects plane *p* in another line parallel (here **HG**) to *l*.) Next, line **H′G′** is parallel to line **HG** because a uniform scale preserves angles. By transitivity **AD** is then parallel to **H′G′**. Hence there is a plane containing **AD** and **H′G′** and lines **AG′** and **DH′** are coplanar. Since coplanar lines intersect, **AG′** and **DH′** intersect. (Note, we are assuming a projective geometry where even parallel lines intersect at their ideal point). The construction defines a mapping on projective 3-space.

On the other hand in the *non*-parallel case the rays **AG′** and **DH′** typically do *not* intersect. By "typically", we mean when $s \neq 1$ and points **H**, **G**, and **O** are not collinear. Figure 5B illustrates the construction when the projection plane and eye axis are not parallel. The proof is by contradiction. Assume lines **AG′** and **DH′** do intersect. Then **AG′** and **DH′** are coplanar and lines **AD** and **H′G′** (dashed red) are also coplanar. Now two coplanar lines are either parallel or intersecting. We stipulated, however, that **AD** (the eye axis) is not parallel to the projection plane, so **AD** cannot be parallel to **H′G′**. Therefore, **H′G′** and **AD** must intersect. We show this cannot be the case either. By construction **AD** and **HG** are coplanar and from the previous paragraph **H′G′** is parallel to **HG**. Therefore line **H′G′** is parallel to plane **ADHG**. But if line **H′G′** is parallel to plane **ADHG**, then line **H′G′** cannot intersect any line in this plane; in particular it cannot intersect **AD**. This contradicts our previous statement that **H′G′** *does* intersect **AD**. Hence **AG′** and **DH′** do not intersect.

This lack of an intersection makes analysis of the complete 3D distortion difficult. However, there are atypical subcases where **AG′** and **DH′** will intersect regardless of eye axis orientation. One such atypical case is when the eyes and the modeled point **E**, are in the XZ plane of Projection Plane Coordinate System. Of course, in this case all projectors are coplanar and hence intersections occur regardless of eye axis orientation. Section 5.2 will analytically

examine this “best”, intersecting case and show that it does not preserve lines. More importantly Section 5.2 will analytically examine the consequences of the more general case when the reconstructing lines do not intersect.

4.2 Image shifting

We now turn our attention to image shifting. Image shifting translates the two projected stereo images toward one and another. In general, this action will both reduce parallax for some modeled points and increase screen parallax for other modeled points. If done carefully this technique can be used to reduce overall maximum screen parallax. Figure 6 illustrates this possibility. In Figure 6A two eye points view three virtual points **A**, **B**, and **C**. The projectors for these points are drawn as black lines distinguished by different line styles. In Figure 6B, image shifting is applied to the projected images of these points. The translation vector is **V** for the left eye image and –**V** for the right eye image. In this example we choose **V** to equal half the screen parallax of point **C**. As a result the new point, **C′**, now has zero screen parallax. Figure 6B shows the effect of this image shifting on all three points. Note that while point **B**’s screen parallax increases, from zero to 2**V**, the overall maximum screen parallax goes down, as indicated by **A′**. Hence image shifting can potentially be used to reduce fusion problems.

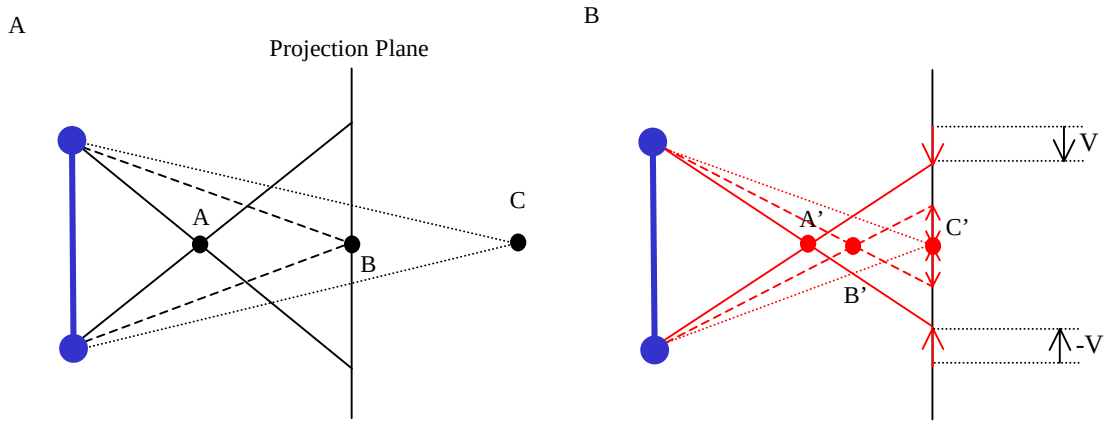


Figure 6: These figures illustrate how image shifting can be used to reduce the global maximum screen parallax of a screen. Figure A contains 3 points, **A**,**B** and **C**, at their modeled position with projectors indicating their left and right eye images. Figure B translates the points' left and right eye images by **V** and –**V** to yield different perceived points, **A′**, **B′** and **C′**.

To understand how image shifting distorts perceived 3D space for a stereo HTD we must be careful about the direction of the translation. An intuitive choice is to translate along the parallel projection of the eye axis onto the projection plane. Figure 7 shows this projected axis in dashed blue. Let T be unit vector on this axis pointing in the direction of the left eye and τ be the magnitude of the desired translation, then the left eye image is translate by vector -τT and the right eye image by τT.

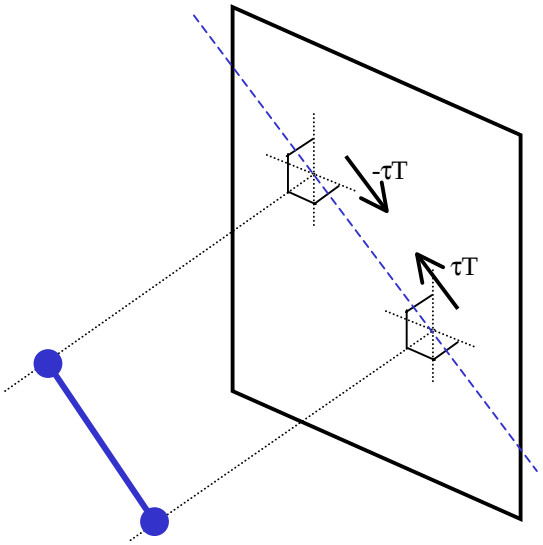


Figure 7: This figure illustrates that the translation vector should be parallel to the perpendicular projection (dash blue) of the eye axis (solid blue).

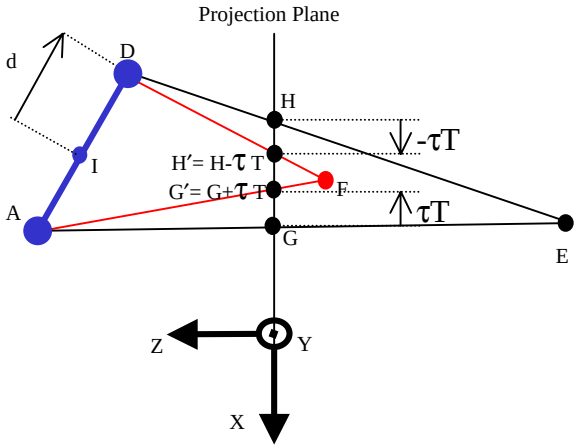


Figure 8: This figure illustrates the construction describing how image shifting maps a modeled point *E* to a perceived point *F*

We can describe the stereoscopic distortion induced by image shifting with the geometric construction illustrated in Figure 8. Figure 8 is a highly abstract diagram of a user viewing a stereoscopic display. The color and labeling conventions follow Figure 4. Modeled point **E** is projected onto the projection plane to points **H** and **G**. Image shifting translates these to points **H′**=**H**-τ**T** and **G′**=**G**+τ**T**. These are the points displayed to the user. The user's visual system reconstructs some perceived point. Ideally this point would be at location F which is the intersection of the lines **AH′** and **DG′**. Unfortunately, the lines **AG′** and **DH′** only generally intersect if the eye axis is parallel to the projection plane. Due to this complication we will analytically examine the parallel case and the non-parallel case separately in Section 5.

First, we verify that in the parallel case lines **AG′** and **DH′** generally will intersect. Since **AD** is parallel to the projection plane, for any point **E** the line **HG** is parallel to the projected eye axis. Since **HG** is parallel to the projected eye axis (dashed blue in Figure 7), **HG** is parallel to **T**. (Recall, we chose **T** to be parallel to the projected eye axis). Therefore H,G, H′ and G′ are always collinear. Since **A** and **D** are coplanar with **H** and **G**, **A** and **D** are also coplanar with **H′** and **G′**. This guarantees the lines **AG′** and **DH′** intersect or are parallel. By treating 3-space as a projective space where parallel lines intersect at a common ideal point, this construction defines a mapping on 3-space. Therefore we can seek an analytic description of the construction for the parallel case.

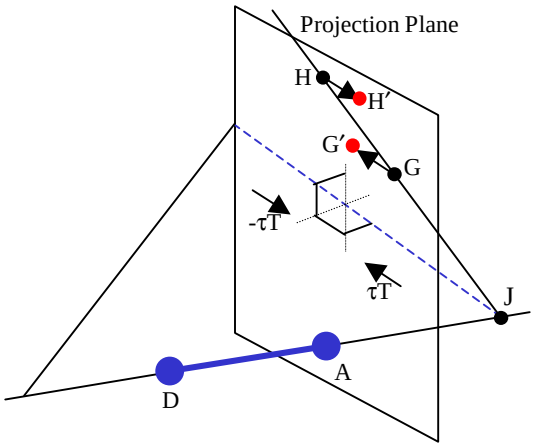


Figure 9: In the non-parallel case image shifting translates homologous points like **H** and **G** by the translation vectors -τT and τT to points **H′** and **G′**. This creates a vertical visual angle displacement, the angle between planes **AJH′** and **AJG′**.

In the non-parallel case, however, **HG** is not generally parallel to the projected eye axis which is the axis along which the 2D images are translated. This is indicated in Figure 9. Consequently, applying the translations -τ**T** and τ**T** to the left and right eye images yields points

not on the line HG. This often results in lines **AG′** and **DH′** being skew. Appendix B Section 2.2 contains a more formal analytic proof of this. This lack of intersection makes it difficult to analytically examine the image shifting distortion. We could handle this problem for image shifting in the same way as we handled it for image scaling. However, this is unnecessary for our purposes because we will show that even the in parallel case, image shifting has additional distortion artifacts beyond α-false eye separation and image scaling.

5 Analytic Descriptions

In this section, we present analytic descriptions of image scaling and image shifting based on the geometric constructions of Section 4. The detailed derivations are found in the appendices. All the results were derived and/or verified using a commercial analysis tool [37]. As mentioned, for both techniques it is necessary to distinguish between the parallel eye axis case and the more general, non-parallel eye axis case.

5.1 Parallel Case

Appendix A Section 1 derives the analytic distortion of image scaling for an eye axis which is parallel to the projection plane. The resulting equation is parameterized on the central eye position, **I**, the vector to the left eye, **d**, and the scale factor *s*, as shown in Figure 4. Using column vector notation the matrix in Projection Plane Coordinates is:

$$\Delta_{sc}^{p} = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & \dfrac{s-1}{Iz} & 1 \end{bmatrix} \tag{3}$$

As a non-affine homology Δ_{sc}^{p} will preserve straight lines but not parallelism. Like α-false eye separation, Δ_{sc}^{p} contains no translation nor dynamic shearing components. However, Δ_{sc}^{p}

does contain a uniform scale component *s* and its effect is clearly evident visually. Neither α-false eye separation nor standard false eye separation have this static scaling component. Δ_{sc}^{p} also varies with head to screen distance, *Iz*, as does α-false eye separation.

Appendix B Section 1 derives an expression for the distortion induced by image shifting assuming the eye axis and projection plane are parallel. The distortion is parameterized on the eye axis center, **I**, the vector to the left eye, **d**, and the translation distance τ. Using column vector notation the matrix in Projection Plane Coordinates is:

$$\Delta_{sh}^{p} = \begin{bmatrix} Q & 0 & -\frac{Ix}{Iz}\tau & Ix\tau \\ 0 & Q & -\frac{Iy}{Iz}\tau & Iy\tau \\ 0 & 0 & Q-\tau & Iz\tau \\ 0 & 0 & -\tau/Iz & \tau+Q \end{bmatrix} \tag{4}$$

where

$$Q = \sqrt{dx^2 + dy^2}$$

Δ_{sh}^{p} is a non-affine homology and hence does not preserve parallelism. Δ_{sh}^{p} contains a translation component in the fourth column. This is to be expected since in Figure 6 we saw that points in the projection plane are moved out of the plane. This can be detrimental for applications which utilize the physical plane of the screen as a work surface for two-dimensional interactions such as precise curve drawing or laying route points on a map. Finally there are dynamic, head position dependent shearing components $-Ix/Iz\ \tau$ and $-Iy/Iz\ \tau$. Neither of these artifacts occur in α-false eye separation nor image scaling.

A pictorial comparison of the distortion artifacts of image scaling, image shifting, false eye separation and α-false eye separation is illustrated in Figure 10 and Figure 11. All these diagrams share the same color coding and format. Figure 10 illustrates how perpendicular (i.e.

perpendicular to the projection plane) head motion effects the perceived 3D image for each fusion control method, while Figure 11 illustrates the effects of lateral head motion. All the diagrams are an overhead view looking down on the user. The projection plane is the middle black line. The eyes are in blue. Instead of illustrating how a single modeled point is mapped to a single perceived point, the diagrams show how an entire modeled grid is mapped to a perceived grid. The modeled grid is black and the perceived grid is red. While the user only sees the perceived grid (red), we overlay it with the modeled black grid for comparison. In A and B, image scaling is applied with scale for of 0.5. In C and D, α-false eye separation is used. The eye separation is underestimated by one-half its true value. The underestimating of the eye separation is illustrated by coloring the true eyes dark blue and modeled eyes light blue. In E and F, standard false eye separation is used. In G and H, image shifting is used with $\tau=0.02$.

Figure 10 illustrates that all four techniques exhibit dynamic artifacts under perpendicular head motion. In contrast, in Figure 11 image scaling (A,B) and α-false eye separation (C,D) do not exhibit dynamic artifacts under lateral head motion but image shifting (G,H) and false eye separation (E,F) do exhibit dynamic artifacts. Keep in mind that since these are 2D diagrams only a side to side head movement is shown, but similar results occur for head movement in any direction parallel to the screen.

In the parallel case, α-false eye separation and image scaling both have fewer dynamic distortion artifacts than image shifting. Image scaling yields dynamic distortion artifacts under the same head motions as α-false eye separation. However, image scaling has a uniform scale component that can drastically change the apparent size of the world and hence the amount of scene seen by the user. In contrast, α-false eye separation has a much smaller effect on the

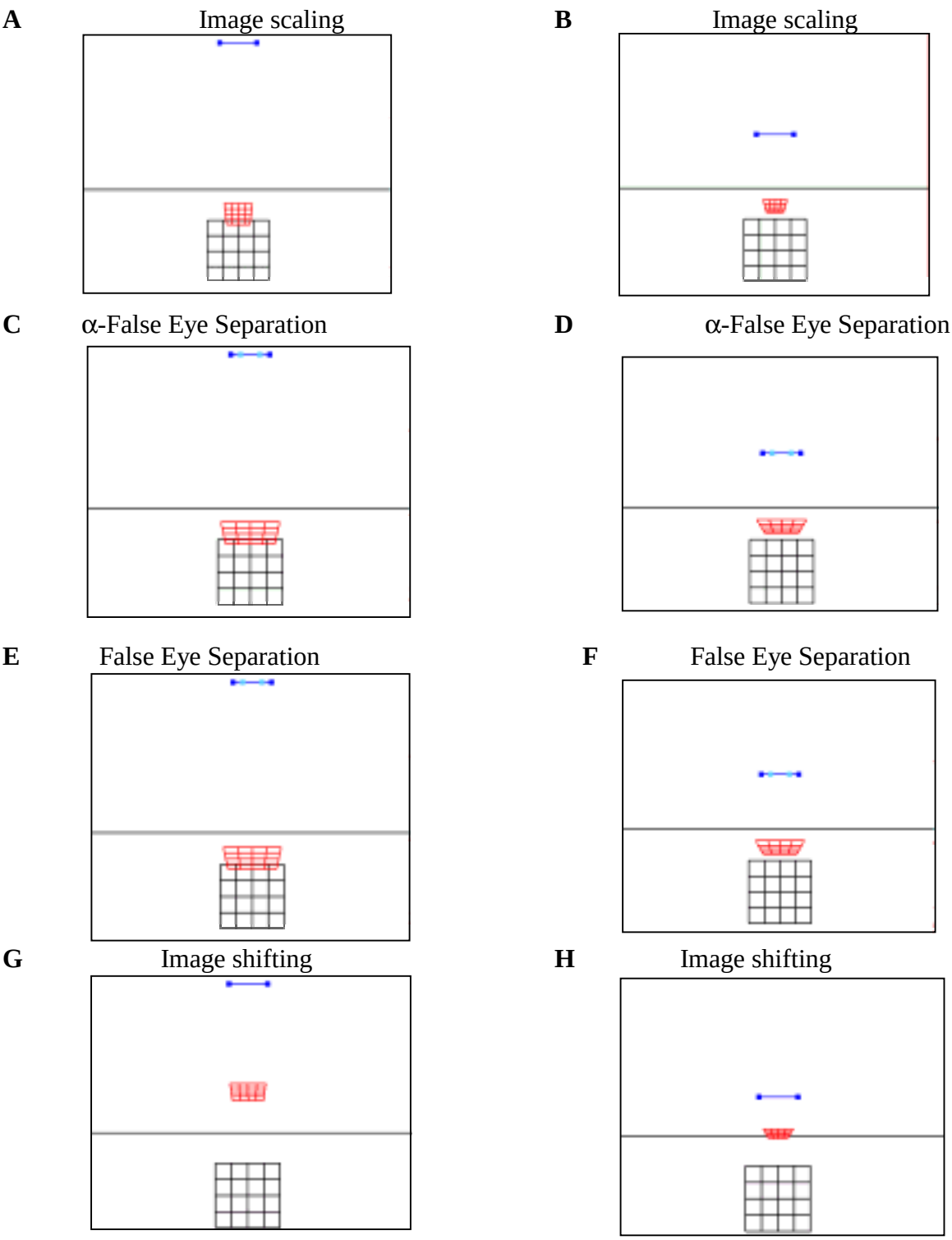
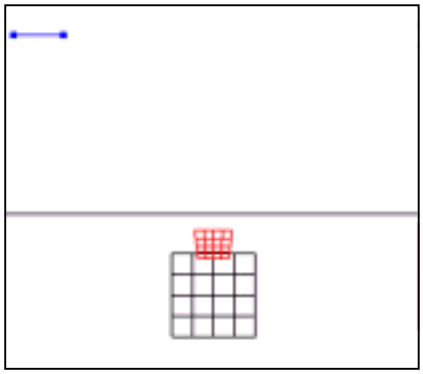
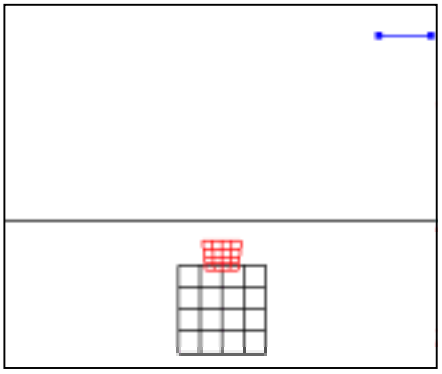


Figure 10: This figure compares the distortion due to forward/backward head movement for several image fusion control methods. (A) and (B) use image scaling with scale factor 0.5. (C) and (D) use α-false eye separation with eye separation ratio 0.5. (E) and (F) use false eye separation with eye separation ratio 0.5. (G) and (H) use image shifting (see Section 5) with translation magnitude 0.02.

A Image scaling

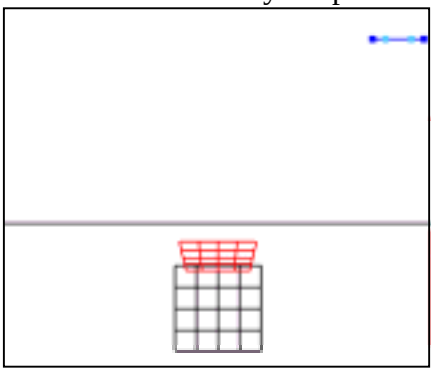


B Image scaling

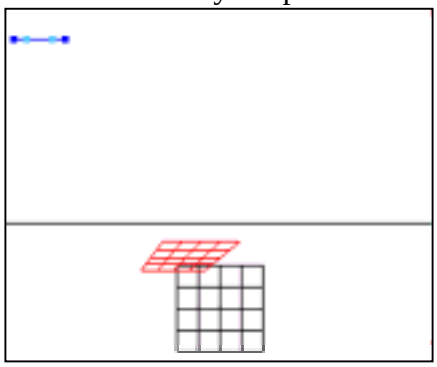


C α-False Eye Separation

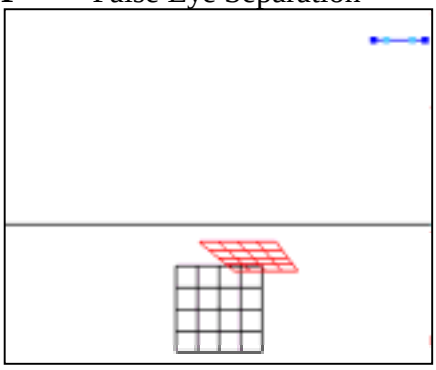
D α-False Eye Separation



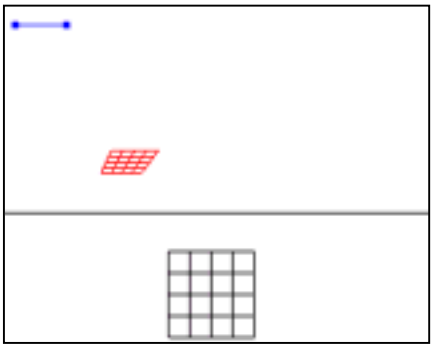
E False Eye Separation



F False Eye Separation



G Image shifting



H Image shifting

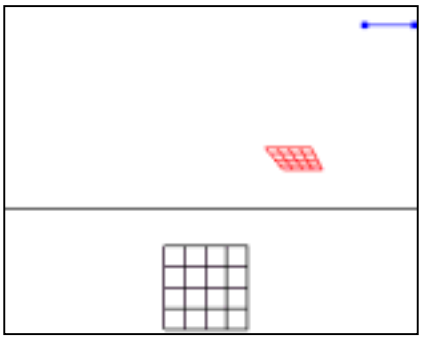


Figure 11: This figure compares the distortion due to left/right head movement for several image fusion control methods. (A) and (B) use image scaling with scale factor 0.5. (C) and (D) use α-false eye separation with eye separation ratio 0.5. (E) and (F) use false eye separation with eye separation ratio 0.5. (G) and (H) use image shifting (see Section 5) with translation magnitude 0.02.

relative field of view and it does not change noticeably. If an application designer desires a fusion control method to alter the perceived scene as subtly as possible then α-false eye separation is a better choice than image scaling. Additionally, if it is important to control the uniform scaling component independently of the view optic adjustment such as in [29] or [33], α-false eye separation is also a better choice since α-false eye separation allows more independent control of the uniform scale factor through the Platform Coordinate System scale. Finally the next section shows that if the eye axis is not parallel to the screen, image scaling has even more serious artifacts not found in α-false eye separation or even standard false eye separation.

5.2 Non-parallel Case

This section considers the analytic distortions in the general case when the eye axis is not parallel to the screen. First we will discuss image scaling in detail. Since image shifting was already shown to yield more distortion artifacts than the other techniques in the parallel case, we will only briefly discuss the image shifting case. As discussed in 5.1 in the non-parallel case for image scaling, key line intersections occur only in special subcases. This section first examines such a special subcase where the modeled point and the eyes are in the XZ plane. Later this section will examine the consequences of the more general subcase.

Appendix A, Section 2 finds the analytic distortion to be:

$$\Delta_{sc}^{XZ}:\ Fx = \frac{\left(\begin{array}{l} Ex\left(dzIx - dxIz\right)\left(dz^2 - Iz^2\right)s + Ex^2\left(dz\,s^2\right)\left(-dz^2 + Iz^2\right) + \\ ExEz\,s\left(-2dzIxIz\left(-1+s\right) + dx\left(-dz^2 - Iz^2 + 2dz^2 s\right)\right) - Ez^2\left(dz\right)\left(dx^2 - Ix^2\right)\left(-1+s\right)s \end{array}\right)}{w} \qquad (5)$$

$$Fz = \left(ExEz\left(-dzs\left(dz^2 - Iz^2\right)\right) + Ez\left(dz^2 - Iz^2\right)\left(dzIx - dxIz\right)s + Ez^2\left(dx\,s\left(dz^2 - Iz^2\right)\right)\right)/\,w$$

$$w = \left(\begin{array}{l} Ex\left(dz\,s\left(-dz^2 + Iz^2\right)\right) + Ez\left(dxIz^2\left(-2+s\right) - 2dzIxIz\left(-1+s\right) + dxdz^2 s\right) + Ez^2\left(dzIx - dxIz\right)\left(-1+s\right) + \\ \left(dzIx - dxIz\right)\left(dz^2 - Iz^2\right) \end{array}\right)$$

Recall **E** and **F** are the modeled and distorted points. **I** is the eye axis center and **d** the vector to the right eye. In the above equation individual coordinates are denoted by appending an 'x', 'y' or 'z'. The scalar *s* is the image scaling factor. The coordinate equations are 2nd degree rational polynomials so image scaling maps lines in modeled space to curves in perceived space. This is shown in Figure 12. Figure 12 is another abstract, overhead view looking down on the user. The eyes are in blue. The projection plane is the black horizontal line. Figure 12 shows the modeled black linear grid (black) is mapped to a curved perceived grid (red).

A **B**

Figure 12: Illustration of the effect on the perceived 3D image when image scaling is applied to an eye axis at an arbitrary orientation in the XZ plane. These diagrams are an overhead view looking down on the user. The diagrams are oriented so that projection plane is horizontal. The

projection plane is the black middle line. The eyes are in blue. The black grid is the modeled grid while the red grid the geometry the user actually perceives. The distortion between the modeled and the perceived space is not a collineation.

While the curvature is quite noticeable in this figure, when viewed from the user's actual point of view on a real stereoscopic HTD the curvature is not noticeable. We could not observe the loss of straight lines when viewing a similar grid on a stereoscopic head-tracked CRT environment with a similar scale factor of 0.5. This is less surprising if we consider that while image scaling distorts 3D space curvilinearly, image scaling only distorts the individual 2D images by a scale which of course preserves lines. So the 2D projected image of a straight line must remain a straight line. Any curvature imparted to the perceived 3D curve exists solely in the depth, or Z components. For example in Figure 13, a line *l* in modeled space is distorted into a perceived curve *l'* by such a distortion; however, the curvature of l'_{xy} is always 0. Humans can perceive a curved surface whose curvature is indicated purely by stereopsis. Julesz [11] illustrates a number of hyperbolic paraboloid and cosine surfaces whose curvature is indicated only by stereopsis cues. However perceiving this curvature probably involves a different mechanism than that used to distinguish curves from straight lines drawn on a piece of paper. Additionally variations in shading and texture provide strong cues to surface shape and in a VR environments these cues are based on the modeled geometry not the stereoscopically distorted (i.e. perceived) geometry. Since we could not observe the induced curvature in a simple wireframe scene when we were explicitly looking for such curvature, we expect that in a more complex, shaded scene a typical user would probably not notice these curvatures either.

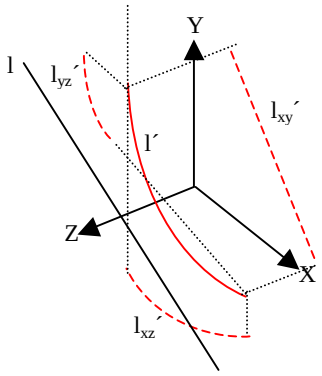


Figure 13: Some of stereo image techniques in this paper map a modeled line *l* to the perceived curve *l´*. However, the straightness of the XY projection of the perceived line *l´* (this projection is l_{XY}´) is preserved. It is the straightness of the XZ and YZ projection (i.e., l_{XZ}´and l_{YZ}´) that is not preserved. The curvature exists in depth only.

What is more troublesome, however, is the fact that the lines drawn between each eye and the corresponding projected and scaled image points do not generally intersect. This was proved in Section 4.1 This means that image scaling displaces homologous points vertically with respect to the eyes. This situation, called vertical parallax, aggravates image fusion problems because humans can fuse only a very small range of vertical parallax [5][11]. Experiments with random dot stereograms show that if vertical parallax is slowly increased until fusion breaks down, the average limit for vertical visual angle (VVA) is only 20 minutes of arc. Additionally, once breakdown does occur VVA must be reduced back to 6 minutes of arc for fusion to reoccur.

We can determine the VVA of the projected and scaled points as shown in Figure 14. **H** and **G** are the projections of a 3D modeled point (not illustrated) onto the projection plane. **H'** and **G'** are **H** and **G** scaled about the window center, **O**. **J** is the intersection of the eye axis **AD** and the projection plane. Note how these points form planes **DJH**, **DJH'**, and **DJG'**. While **H** and

G form one common plane, **H'** and **G'** for two more different planes. The VVA separating **H'** and **G'** is the angle between planes **DJH'** and **DJG'** as measured about the line **AD**.

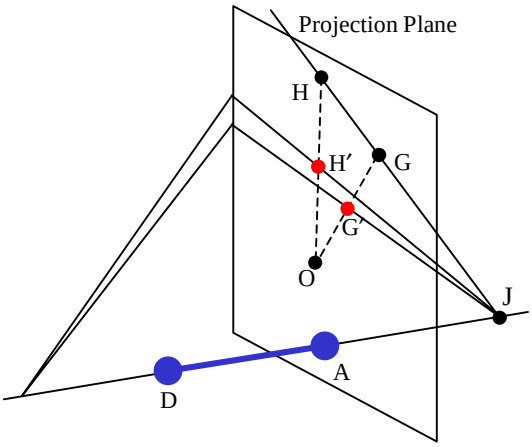


Figure 14: Illustration of how image scaling yields vertical parallax. **H** and **G** are the left and right eye images of a modeled point. Image scaling scales them to points **H'** and **G'**. **H'** and **G'** are displaced by the vertical visual angle measured between planes **DJG'** and **DJH'**.

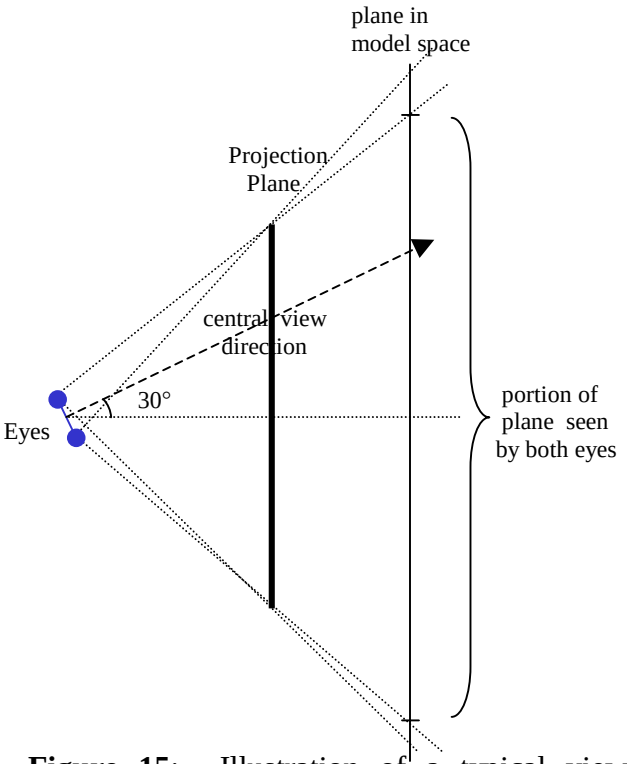


Figure 15: Illustration of a typical view configuration used to create a 3D plot of VVA for a given plane in model space.

Using the above construction we can compute the VVA for various points in modeled space given a specific viewing configuration and image scaling scale factor. We can graph the VVA

for a set of model points contained in some plane with a fixed z coordinate. Figure 15 illustrates the idea. We pick a plane in model space with a fixed z-coordinate. We compute the x and y extents of the square window in this plane that is viewable from both the left and right eye. In Figure 15, this area is delimited by horizontal marks on the model plane. We project each point in this plane to the projection plane once for each eye. We then scale the two projected points via image scaling and calculate the VVA value. We plot the data as a 3D graph with X and Y on two axes and VVA on the third axis. Since stereopsis only occurs over a limited field of view, the VVA values for the model points that are directly in front of the eyes are of primary importance. This is indicated in Figure 15 by the dash line marked "central view direction."

We explored such 3D plots for a variety of screen sizes, eye axis angles, eye axis to screen distances, and model plane depths. A typical plot is shown in Figure 16. The plot is shaped like a curved 'V' extruded along the x-axis. This plot is for an upright 2 x 2 meter screen. The eye separation is 0.06 m. The axis is twisted 30 degrees looking towards the left of the screen while remaining in the XZ plane. The axis center is located at (0,0.524,1) in projection plane coordinates. (Figure 15 shows an overhead view of a similar situation). This accounts for a 5.5 foot tall user standing 1 meter from the screen with eyes 5 feet off the ground. The model plane is at z=-100 (i.e. 100 meters behind the screen). The scale factor is 0.527. This factor is used so as to bring the maximum possible uncrossed horizontal parallax of 0.06 m to within the 1.57 HVA limit discussed in Section 2. At the given line of sight distance of 1.154 m (1/cos30) from the screen, this 1.57 HVA translates to a horizontal parallax of 0.0316 m. The shape of Figure 16 is typical of these types of plots. VVA grows larger as the y coordinate moves away from the valley of the 'V' and VVA tends to shrink as the x-coordinate moves in the direction of

head twist. In Figure 16, the user is looking to the left and the VVA shrinks as the x-coordinate moves to the left. This plot shows VVA meeting or exceeding the 20 arc minute limit [5][11] for some y-position for every x-position. The 6 arc minute re-fusion limit is far exceeded for an even larger portion of the space. Importantly, the resulting VVA is displayable in pixels. Assume the screen is 1000 by 1000 so that pixels are 0.002 meters tall. Along the central view direction the VVA reaches 22′ at the top of the screen. At this point on the screen a 22' VVA spans 0.0128 meters or 6 pixels. Hence, the VVA is significant pixel-wise. In this example by making the HVA acceptable, we've made the VVA unacceptable!

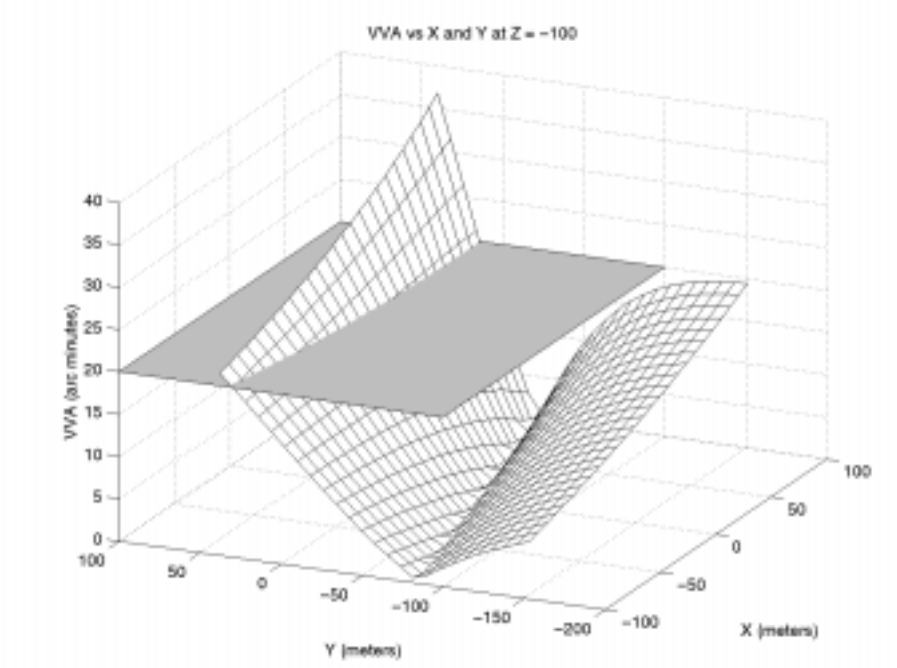


Figure 16: A plot of VVA for a plane of points in space at Z=-100.

Ideally analysis would show either that image scaling yields acceptable VVA for all display configurations or that image scaling yields unacceptable VVA for all configurations. Unfortunately, this is not the case because VVA depends on many factors. There are a large

number of independent variables including screen size, eye axis orientation, eye axis position, and model plane depth. Second, there are algorithmic variations. An image scaling algorithm can use either a fixed scale factor or a variable scale factor. A variable scale factor could be chosen to either avoid crossed diplopia or uncrossed diplopia or both. All these factors would have to be considered by an application designer in order to determine whether image scaling will exceed VVA limits for a particular application with a particular display configuration. Contrast this situation with that of α-false eye separation (or even false eye separation). α-false eye separation never creates any vertical parallax at all. This follows easily from the construction of regular false eye separation and the fact that the α predistortion, a 3D transformation, is applied to virtual space as a whole and not to the individual left and right eye images. Since image scaling has no advantages in terms of performance or distortion artifacts and since only α-false eye separation avoids vertical parallax, we conclude that α-false eye separation is the better choice for general purpose image fusion control for stereo HTD's.

In regards to image shifting, section 5.1 showed that image shifting has additional distortion artifacts beyond both image scaling and α-false eye separation for the parallel case. Therefore a full investigation of the non-parallel case is not necessary in order to establish a ranking among these techniques. We only mention that like image scaling, in the non-parallel case image shifting typically yields vertical parallax. In special subcases where the reconstructing rays do intersect, image shifting also distorts the perceived image by a different 2^{nd} degree rational polynomial (Appendix B, Section 2.1).

6 Conclusions and Future Work

In this paper we have shown that for single screen HTD's where the eye axis is parallel to the projection plane, α-false eye separation and image scaling are better choices than image shifting in terms of 3D distortion artifacts. In particular the latter two methods do not exhibit a translation component nor dynamic left/right shearing. If a practitioner desires either to make the image fusion distortion as subtle as possible or to control the uniform scaling factor independently of the image fusion parameter, α-false eye separation must be used instead of image scaling. This is because the image scaling distortion mixes in an inherent uniform scale component. Finally, for general HTD configurations where the eye axis is not parallel to the displays, α-false eye separation is the best choice since only α-false eye separation avoids vertical parallax and subtle curvature effects.

Our next major step is to perform usability testing and application analysis to compare α-false eye separation to the direct 3D techniques of optic scaling and viewer scaling. These comparisons will be made for a variety of applications parameterized on several attributes. These attributes will include: (1) whether the application uses the near space (the space in front of the projection plane), the far space (the space beyond the projection plane) or both spaces; (2) how scene geometry is distributed over this space—for instance is it uniformly distributed in depth or does it occur in clumps; and (3) how the user interacts with the application—is it just a fly through or does the user grab and manipulate nearby objects. Preliminary results indicate that these factors greatly affect which technique is preferable. We are also investigating direct transform techniques that improve upon α-false eye separation by removing all dynamic distortion artifacts and controlling near and far fusion problems simultaneously.

References

[1] C. Cruz-Neira, D.J. Sandin, T.A. DeFanti, "Surround-screen projection-based virtual reality: the design and implementation of the CAVE," *SIGGRAPH 93 Conference Proceedings, Annual Conference Series*, ACM SIGGRAPH: Addison Wesley, Anaheim, CA, August 1993, pp. 135-42.

[2] J.E. Cutting, "How the eye measures reality and virtual reality," *Behavioral Research Methods, Instruments & Computers*, Vol. 29, No. 1, Feb. 1997, pp. 27-36.

[3] E.T. Davis and L.F. Hodges, "Human Stereopsis, Fusion, and Stereoscopic Virtual Environments," *Virtual Environments and Advanced Interface Design*, Woodrow Barfield and Thomas A. Furness III, ed., New York/Oxford: Oxford University Press, 1995, pp. 145-175.

[4] M. Deering, "High Resolution Virtual Reality," *Computer Graphics (SIGGRAPH 92 Conference Proceedings)*, Vol. 26, July 1992, pp. 195-202.

[5] D. Fender, B. Julesz, "Extension of Panum's Fusional Area in Binocularly Stabilized Vision," *Journal of the Optical Society of America*, Vol. 57, No. 6, June 1967, pp. 819-831.

[6] J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Huges. *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Company. 1992.

[7] Larry F. Hodges, David F. McAllister. Rotation algorithm artifacts in stereoscopic images. *Optical Engineering*. 29(8): 973-976, August 1990.

[8] L.F. Hodges, “Tutorial: Time-Multiplexed Stereoscopic Computer Graphics,” *IEEE Computer Graphics and Applications,* Vol.12, No.2, March 1992, pp.20-30.

[9] L.F. Hodges, E.T. Davis, “Geometric Considerations for Stereoscopic Virtual Environments,” *Presence,* Vol. 2, No 1, Winter 1993, pp. 34-43.

[10] L.F. Hodges, D. McAllister. Computing Stereoscopic Views. In David McAllister, editor, *Stereo Computer Graphics and Other True 3D Technologies.* Princeton University Press-Princeton, New Jersey. 1993.

[11] B. Julesz, *Foundations of Cyclopean Perception,* The University of Chicago Press. Chicago and London, 1971.

[12] J. Konrad, ”Enhancement of viewer comfort in stereoscopic viewing: parallax adjustment,” *IS&T/SPIE Symposium on Electronic Imaging, Stereoscopic Displays and Virtual Reality Systems,* San Jose, CA, 1999, pp. 179-190.

[13] W. Krüger. C.-A. Bohn, B. Fröhlich, H. Schüth, W. Strauss, G. Wesche, “The Responsive Workbench: A Virtual Work Environment,” *IEEE Computer*, Vol. 28. No. 7. July 1995. pp 42-48.

[14] L. Lipton, *Foundations of the Stereoscopic Cinema: A Study in Depth,* Van Nostrand Reinhold, 1982.

[15] D. McAllister, “Introduction,” *Stereo Computer Graphics and Other True 3D Technologies,* David McAllister, ed., Princeton University Press-Princeton, New Jersey. 1993.

[16] M. Mon-Williams, J. P.Wann, and S. Rushton, “Design Factors in stereoscopic virtual-reality displays,” *Journal of the SID,*Vol. 3, No. 4, Dec. 1995, pp. 207-210.

[17] W. Robinett, J.P. Rolland, “A Computational Model for the Stereoscopic Optics of a Head-Mounted Display,” *Presence*, Vol. 1, No. 1, Winter 1992. pp 45-62.

[18] W. Robinett, R. Holloway, “The Visual Display Transformation for Virtual Reality,” *Presence*, Vol. 4, No. 1, Winter 1995, pp. 1-23.

[19] J.P. Rolland and W. Gibson, “Toward Quantifying Depth and Size Perception in Virtual Environments,” *Presence*, Vol 4., No 1., Winter 1995, pp. 24-49.

[20] L.B. Rosenberg, “The Effect of Interocular Distance upon Operator Performance using Stereoscopic Displays to Perform Virtual Depth Tasks,” *IEEE Virtual Reality Annual International Symposium 93*, Seattle, WA, 1993, pp. 27-32.

[21] M. Siegel, Y. Tobinaga, and T. Akiya, “Kinder Gentler Stereo,” *IS&T/SPIE Symposium on Electronic Imaging, Stereoscopic Displays and Virtual Reality Systems*, San Jose, CA, 1999, pp. 18-27.

[22] D.A. Southard, “Transformations for Stereoscopic Visual Simulation,” *Computer & Graphics*, Vol. 16, No. 4, Winter 1992, pp.401-410.

[23] D.A. Southard, “Viewing Model for Virtual Environment Displays,” *Journal of Electronic Imaging*, Vol. 4, No. 4, Oct. 1995, pp 413–420.

[24] R.T. Surdick, E.T. Davis, R.A. King, L.F. Hodges, “The Perception of Distance in Simulated Displays,” *Presence*, Vol. 6, No. 5, Oct. 1997, pp. 513-531.

[25] C. W. Tyler, “Induced Stereomovement,” *Vision Research*, Vol. 14, August 1974, pp. 609-613.

[26] N. A. Valyus, *Stereoscopy*, The Focal Press, London and New York, 1966.

[27] C. Ware, K. Arthur and K.S. Booth, “Fish Tank Virtual Reality,” In proceedings of *InterChi ’93*, Apr. 1993, pp. 37-41.

[28] C. Ware, C. Gobrecht and M. Paton, “Algorithm for dynamic disparity Adjustment,” *Proceedings of the SPIE - The International Society for Optical Engineering*. *Stereoscopic Displays and Virtual Reality Systems II*, Vol. 2409, Feb. 1995, pp. 150-156.

[29] C. Ware, “Dynamic Stereo Displays,” *CHI’95 Mosaic of Creativity*, ACM Press: Addison Wesley, Denver, CO, 1995, pp. 310-316.

[30] C. Ware, and D. Fleet, “Integrating Flying and Fish Tank Metaphors with Cyclopean Scale,” Proceedings of Computer Graphics International, Hasselt and Diepenbeek, Belgium, 23-27 June 1997, pp.39-46.

[31] C. Ware, C. Gobrecht, and M.A. Paton, “Dynamic Adjustment of Stereo Display Parameters,” *IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans*, Vol. 28, No. 1, Jan. 1998, pp. 56-65.

[32] Z. Wartell, L. Hodges, and W. Ribarksy. “The Analytic Distortion Induced by False-Eye Separation in Head-Tracked Stereoscopic Displays,” GVU Tech Report 99-01, Computer Science Dept., Georgia Institute of Technology, Atlanta, GA, 1999.

[33] Z. Wartell, W. Ribarsky, and L. Hodges, "Third-Person Navigation of Whole-Planet Terrain in a Head-Tracked Stereoscopic Environment," *Proceedings of IEEE Virtual Reality '99 Conference*, IEEE Computer Society Press, Houston, Texas, 1999, pp. 141-148.

[34] Z. Wartell, L.F. Hodges, and W. Ribarsky, "Balancing Fusion, Image Depth, and Distortion in Stereoscopic Head-Tracked Displays," *SIGGRAPH 99 Conference Proceedings, Annual Conference Series*, ACM SIGGRAPH, Addison Wesley, Los Angeles, CA, 1999, pp. 351-357.

[35] Z. Wartell, L.F. Hodges, and W. Ribarsky. "An Analytic Comparison of α-False Eye Separation, Image scaling and Image shifting in Stereoscopic Displays," GVU Tech Report 99-XX, Computer Science Dept., Georgia Institute of Technology, Atlanta, GA, 1999.

[36] S.P. Williams and R. V. Parrish, "New computational control techniques and increased understanding for stereo 3-D displays," *Stereoscopic Displays and Applications*, Proceedings of the SPIE - The International Society for Optical Engineering, Santa Clara, CA, 1990, pp. 73-82.

[37] Mathematica, Wolfram Research Inc., 1988-1996.

[38] A. Woods, T. Docherty, R. Koch, "Image Distortion in Stereoscopic Video Systems," *Proceedings of the SPIE – The International Society for Optical Engineering, Stereoscopic Displays and Applications IV*, Vol. 1915, 1993, pp. 36 – 48.

[39] Y.-Y. Yeh and L.D. Silverstein, "Limits of Fusion and Depth Judgements in Stereoscopic Color Displays," *Human Factors*, Vol. 32, No. 1, Feb. 1990, pp. 45-60.

APPENDIX A: IMAGE SCALING DISTORTION

1 Parallel Case

The following figure illustrates the distortion induced by image scaling for a head at an arbitrary position but oriented parallel to the projection plane. The eye points are on the left in blue and the projection plane is in the X-Y plane. The projection plane is in the XY plane and the projection window is centered about the origin. (Note, Figure A-1 only shows a portion of the projection window so the window does not actually appear centered in the diagram). *E* is the modeled object point and *F* (red) is the perceived object point. The user's central eye point is at *I*. The left eye, *D*, is displaced by *d* from *I* and the right eye, *A*, is displaced by *-d*. 2|*d*| is the true eye separation. We assume correct modeling of the eye separation. *E* is projected onto the points *H* and *G* on the projection plane. Image scaling by scalar factor *s* scales points *H* and *G* into points *s*·*H* and *s*·*G*. These scaled points are those the user sees. The user reconstructs the point at location *F*. Note for this parallel case *A*,*D*,*d* and *E* are not restricted to be in the XZ plane.

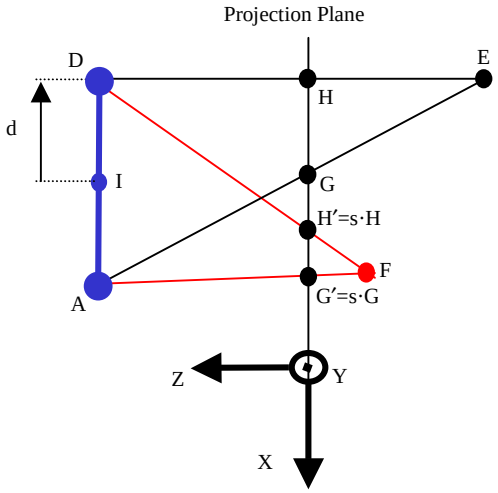


Figure A-1: This is an abstract diagram of a user viewing a stereoscopic HTD at a parallel orientation. See above text for details.

1.1 From Figure A-1:

A = I – d
D = I + d
dz=0

1.2 Solve for H:
Equation of line DH is:
P = (E-D)t + D

At z = 0:

$$t = \frac{Dz}{Dz - Ez}$$

So:

$$H = (E - D)\frac{Dz}{Dz - Ez} + D$$

Or from 1.1:

$$H = (E - I - d)\frac{Iz + dz}{Iz - Ez} + I + d$$

1.3 Solve for G:

Using arguments similar to 1.2:

$$G = (E - A)\frac{Az}{Az - Ez} + A = (E - I + d)\frac{Iz - dz}{Iz - Ez} + I - d$$

1.4 Solve for Fx:

To begin:

$$F = \overline{AsG} \cap \overline{DsH}$$

In [32] we derived a similar result for distortions due to false eye separation. By substituting $s \cdot G$ for the G and $s \cdot H$ for the H for in the final result of Appendix 1.4 in [32] we have

$$Fx = \frac{AzDx\, s\, Gx - AxDz\, s\, Hx + (Dz - Az)\, s\, Gx\, s\, Hx}{AzDx - AxDz + Dz\, s\, Gx - Az\, s\, Hx}$$

Our previous manual derivation for this type of expression was quite laborous and tedious [32], therefore we use a commercial software algebra solver. To find *Fx*, we use Mathematica [37] with the following input file:

```
Hx=(Ex-Ix-dx)(Iz+dz)/(Iz-Ez+dz)+Ix+dx
Gx=(Ex-Ix+dx)(Iz-dz)/(Iz-Ez-dz)+Ix-dx

Dx = Ix + dx
Dz = Iz + dz
Ax = Ix - dx
Az = Iz - dz
dz = 0

(* FX *)
Num=Expand[Az Dx s Gx - Ax Dz s Hx + (Dz-Az)s Gx s Hx]
Num = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Num]
Num = Factor[Num]

Den = Expand[Az Dx - Ax Dz + Dz s Gx - Az s Hx]
Den = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Den]
Den = Factor[Den]

Fx = Cancel[Num/Den]
Fx = Collect[Numerator[Fx],{Ez,Ex}]/Collect[Denominator[Fx],{Ex,Ez}]
```

The final result is:

$$Fx = \frac{ExIz \quad s}{Ez(s-1) + Iz}$$

1.5 Solve for Fy:

Solving for *Fy* uses a parallel derivation to *Fx*, yielding:

$$Fy = \frac{EyIz \quad s}{Ez(s-1) + Iz}$$

1.6 Solve for Fz:

Using the initial results from Appendix 1.6 in [32] and substituting *s H* for *H* and *s G* for *G*:

$$Fz = \frac{AzDzsHx - AzDzsGx}{-AzDx + AxDz - DzsGx + AzsHx}$$

Expression for H and G are in 1.2 and 1.3 of this appendix. To find *Fz*, we use Mathematica [37] with the following input file:

```
Hx=(Ex-Ix-dx)(Iz+dz)/(Iz-Ez+dz)+Ix+dx
Gx=(Ex-Ix+dx)(Iz-dz)/(Iz-Ez-dz)+Ix-dx

Dx = Ix + dx
Dz = Iz + dz
Ax = Ix - dx
Az = Iz - dz
dz = 0

(* FZ *)
Num = Expand[Az Dz s Hx - Az Dz s Gx]
Num = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Num]
Num = Factor[Num]

Den = Expand[-Az Dx + Ax Dz - Dz s Gx + Az s Hx]
Den = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Den]
Den = Factor[Den]

Fz = Cancel[Num/Den]
Fz = Collect[Numerator[Fz],{Ez}]/Collect[Denominator[Fz],{Ez}]
```

This yields:

$$Fz = \frac{Ez\,Iz \quad s}{Ez(s-1) + Iz}$$

1.7 Rewrite in Matrix Form:

Combining 1.4-1.6 and using column vector notation the distortion matrix is:

$$\Delta_{sc}^{p} = \begin{bmatrix} Izs & 0 & 0 & 0 \\ 0 & Izs & 0 & 0 \\ 0 & 0 & Izs & 0 \\ 0 & 0 & s-1 & Iz \end{bmatrix} = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & \frac{s-1}{Iz} & 1 \end{bmatrix}$$

The last simplification is possible since homology matrices are only unique up to a scale factor and we assume the eye axis center is not embedded in the projection plane.

2 Non-Parallel Case

The following figure illustrates the distortion induced by image scaling for a head at an arbitrary position and orientation. The eye points are on the left in blue and the projection plane is in the X-Y plane. The projection plane is in the XY plane and the projection window is centered about the origin. (Note, Figure A-2 only shows a portion of the projection window so the window does not actually appear centered in the diagram). *E* is the modeled object point and *F* (red) is the perceived object point. The user's central eye point is at *I*. The left eye, *D*, is displaced by *d* from *I* and the right eye, *A*, is displaced by *-d*. 2|*d*| is the true eye separation. We assume correct modeling of the eye separation. *E* is projected onto the points *H* and *G* on the projection plane. Image scaling by scalar factor *s* scales points *H* and *G* into points *H′=s·H* and *G′=s·G*. These scaled points are those the user sees. As discussed in Section 4.1, lines DH′ and AG′ do not generally intersect. Therefore, we restrict ourselves to a simple special case where they do intersect. *A*,*D*,*d* and *E* are restricted to be in the XZ plane as shown in Figure A-2. Since *Ay*,*Dy*,*dy*,*Ey* equal zero and the y coordinates of all dependent points, *H*, *G*, etc., are also zero. This would yield a well-define point at location *F* as discussed in Section 4.1 of the text.

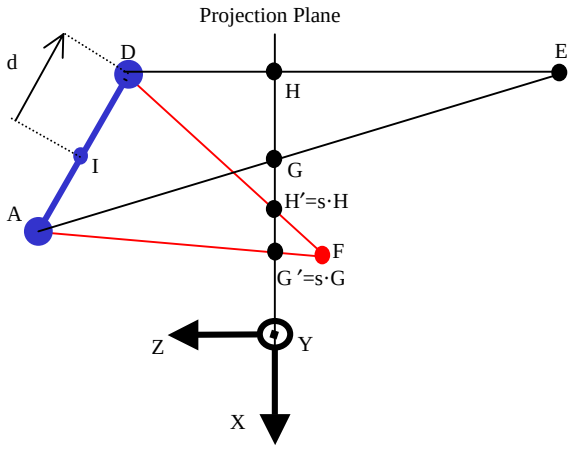


Figure A-2: This is an abstract diagram of a user viewing a stereoscopic HTD at an arbitrary orientation. See above text for details.

The derivation is parallel to Appendix A Section 1. When computing the expressions for *Fx* and *Fz* just remove the line 'dz=0' in the respective Mathematica files (Appendix A 1.4 and Appendix A 1.6). *Fy* is simply 0. These alterations account for the fact that we are limiting ourselves to the XZ plane and that the eye axis is no longer parallel to the screen. The resulting expression is:

$$\Delta_{sc}^{XZ}:\ Fx = \left(\begin{array}{l} Ex\left(dzIx - dxIz\right)\left(dz^2 - Iz^2\right)s + Ex^2\left(dz\,s^2\right)\left(-dz^2 + Iz^2\right) + \\ ExEz\,s\left(-2dzIxIz\left(-1+s\right) + dx\left(-dz^2 - Iz^2 + 2dz^2 s\right)\right) - Ez^2\left(dz\right)\left(dx^2 - Ix^2\right)\left(-1+s\right)s \end{array} \right) / w$$

$$Fz = \left(ExEz\left(-dzs\left(dz^2 - Iz^2\right)\right) + Ez\left(dz^2 - Iz^2\right)\left(dzIx - dxIz\right)s + Ez^2\left(dx\,s\left(dz^2 - Iz^2\right)\right)\right) / w$$

$$w = \left(\begin{array}{l} Ex\left(dz\,s\left(-dz^2 + Iz^2\right)\right) + Ez\left(dxIz^2\left(-2+s\right) - 2dzIxIz\left(-1+s\right) + dxdz^2 s\right) + Ez^2\left(dzIx - dxIz\right)\left(-1+s\right) + \\ \left(dzIx - dxIz\right)\left(dz^2 - Iz^2\right) \end{array} \right)$$

APPENDIX B: IMAGE SHIFTING DISTORTION

1. Parallel Case

The following figure illustrates the distortion induced by image shifting for a head at an arbitrary position but oriented parallel to the projection plane. The eye points are on the left in blue and the projection plane is in the X-Y plane. The projection plane is in the XY plane and the projection window is centered about the origin. (Note, Figure B-1 only shows a portion of the projection window so the window does not actually appear centered in the diagram). *E* is the modeled object point and *F* (red) is the perceived object point. The user's central eye point is at *I*. The left eye, *D*, is displaced by *d* from *I* and the right eye, *A*, is displaced by *-d*. 2|*d*| is the true eye separation. We assume correct modeling of the eye separation. *E* is projected onto the points *H* and *G* on the projection plane. Window translates points *H* and *G* by distance τ along vector T which is the XY-projected unit vector of *d*. This yields $H-\tau T$ and $G+\tau T$ which are seen by the user. The user reconstructs the point at location *F*.

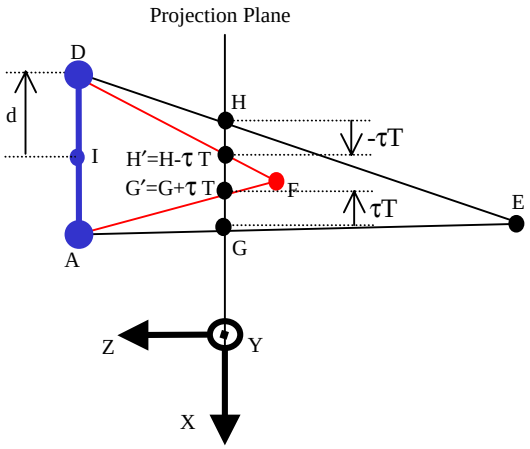


Figure B-1: This is an abstract diagram of a user viewing a stereoscopic HTD a parallel orientation. See above text for details.

1.1 From the figure:

A = I – d
D = I + d
dz=0
T = (dx/sqrt(dx^2+ dy^2), dy/sqrt(dx^2+ dy^2))

1.2 Solve for Fx:
To begin:

$$F = \overline{AG'} \cap \overline{DH'}$$

where

$$G' = G + \tau T$$

$$H' = H - \tau T$$

In [32] we derived a similar result for distortions due to false eye separation modeling. By substituting *G'* for the *G* and *H'* for the *H* in the final result of Appendix 1.4 in [32] we have

$$Fx = \frac{AzDx\ Gx' - AxDz\ Hx' + (Dz - Az)\ Gx'\ Hx'}{AzDx - AxDz + Dz\ Gx' - Az\ Hx'}$$

Expressions for *H* and *G* are in appendix A-1.2 and A-1.3. Manual derivations for this type expression is quite tedious [32], therefore we use Mathematica [37] with the following input file:

```
Clear["*"]

Hx=(Ex-Ix-dx)(Iz+dz)/(Iz-Ez+dz)+Ix+dx
Gx=(Ex-Ix+dx)(Iz-dz)/(Iz-Ez-dz)+Ix-dx

Tx = dx / Sqrt[dx*dx + dy*dy]

Hx = Hx - tau Tx
Gx = Gx + tau Tx

Dx = Ix + dx
Dz = Iz + dz
Ax = Ix - dx
Az = Iz - dz
dz = 0

(* FX *)
Num=Expand[Az Dx Gx - Ax Dz Hx + (Dz-Az) Gx Hx]
Num = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Num]
Num = Factor[Num]

Den = Expand[Az Dx - Ax Dz + Dz  Gx - Az Hx]
Den = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Den]
Den = Factor[Den]

Fx = Cancel[Num/Den]
Fx = Collect[Numerator[Fx],{Ex,Ez}]/Collect[Denominator[Fx],{Ex,Ez}]
```

The final result is:

$$Fx = \frac{Ex\,Iz\sqrt{dx^2 + dy^2} - Ez\,Ix\,\tau + IxIz\,\tau}{-Ez\,\tau + Iz\left(\tau + \sqrt{dx^2 + dy^2}\right)}$$

1.3 Solve for Fy:

Solving for *Fy* uses a parallel derivation to *Fx*, yielding:

$$Fy = \frac{Ey\,Iz\sqrt{dx^2 + dy^2} - Ez\,Iy\,\tau + IyIz\,\tau}{-Ez\,\tau + Iz\left(\tau + \sqrt{dx^2 + dy^2}\right)}$$

1.4 Solve for Fz:

Substituting H' for H and G' for G in results from Appendix 1.6 in [32]:

$$Fz = \frac{AzDz\,Hx' - AzDz\,Gx'}{-AzDx + AxDz - Dz\,Gx' + Az\,Hx'}$$

Next we use Mathematica with the following input file:

```
Clear["*"]

Hx=(Ex-Ix-dx)(Iz+dz)/(Iz-Ez+dz)+Ix+dx
Gx=(Ex-Ix+dx)(Iz-dz)/(Iz-Ez-dz)+Ix-dx

Tx = dx / Sqrt[dx*dx + dy*dy]

Hx = Hx - tau Tx
Gx = Gx + tau Tx

Dx = Ix + dx
Dz = Iz + dz
Ax = Ix - dx
Az = Iz - dz
dz = 0


(* FZ *)
Num = Expand[Az Dz  Hx - Az Dz Gx]
Num = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Num]
Num = Factor[Num]

Den = Expand[-Az Dx + Ax Dz - Dz  Gx + Az  Hx]
Den = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Den]
Den = Factor[Den]

Fz = Cancel[Num/Den]
Fz = Collect[Numerator[Fz],{Ez}]/Collect[Denominator[Fz],{Ez}]
```

This yields:

$$Fz = \frac{Ez\,Iz\left(\sqrt{dx^2 + dy^2} - \tau\right) + Iz^2\tau}{-Ez\,\tau + Iz\left(\tau + \sqrt{dx^2 + dy^2}\right)}$$

1.5 Rewrite in Matrix Form:

Combining 1.2-1.4 and using a column vector notation the matrix the is:

$$\Delta_{sh}^{p} = \begin{bmatrix} Iz\,Q & 0 & -Ix\,\tau & IxIz\,\tau \\ 0 & IzQ & -Iy\,\tau & IyIz\,\tau \\ 0 & 0 & IzQ - Iz\,\tau & Iz^2\tau \\ 0 & 0 & -\tau & Iz(\tau + Q) \end{bmatrix} = \begin{bmatrix} Q & 0 & -\frac{Ix}{Iz}\tau & Ix\,\tau \\ 0 & Q & -\frac{Iy}{Iz}\tau & Iy\,\tau \\ 0 & 0 & Q - \tau & Iz\,\tau \\ 0 & 0 & -\tau/Iz & \tau + Q \end{bmatrix}$$

where

$$Q = \sqrt{dx^2 + dy^2}$$

The final simplification is possible since homology matrices are only unique up to a scale factor and we assume the eye axis center is not embedded in the projection plane.

2 Non-Parallel Case

Section 2.1 discusses the analytic expression for image shifting in the non-parallel case if we restrict the eyes and the modeled point *E* to the XZ plane. 2.2 shows that without this restriction the reconstructing projectors do not generally intersect.

The following figure illustrates the distortion induced by window scaling for a head at an arbitrary position and orientation. The eye points are on the left in blue and the projection plane is in the X-Y plane. The projection plane is in the XY plane and the projection window is centered about the origin. (Note, Figure B-2 only shows a portion of the projection window so the window does not actually appear centered in the diagram). *E* is the modeled object point and *F* (red) is the perceived object point. The user's central eye point is at *I*. The left eye, *D*, is displaced by *d* from *I* and the right eye, *A*, is displaced by *-d*. $2|d|$ is the true eye separation. We assume correct modeling of the eye separation. *E* is projected onto the points *H* and *G* on the projection plane. Window translates points *H* and *G* by distance τ along vector T which is the XY-projected unit vector of *d*. This yields $H\text{-}\tau T$ and $G+\tau T$ which are seen by the user. As discussed earlier lines (*D*, $H\text{-}\tau T$) and (*A*, $G+\tau T$) do not generally intersect. Therefore, we restrict ourselves to a simple special case where they do intersect. We restrict A,D,d and E to be contained in the XZ plane. So Ay, Dy, dy, Ey equal zero and hence the y-coordinates of all dependent points are also zero. The user reconstructs the point at location *F*.

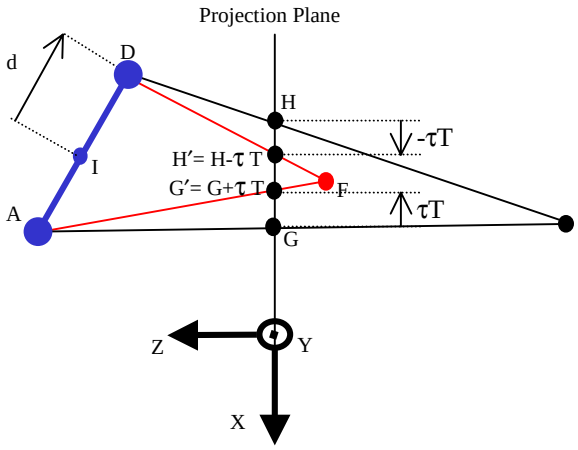


Figure B-2: This is an abstract diagram of a user viewing a stereoscopic HTD at an arbitrary orientation. See above text for details.

2.1 Analytic Expression

The derivation is parallel to Appendix B Section 1. When computing the expressions for *Fx* and *Fz* just replace 'dy=0' for 'dz=0' in line 15 of the respective Mathematica files (Appendix B 1.2 and Appendix B 1.4). *Fy* is just 0. These alterations account for the fact that we are limiting ourselves to the XZ plane and that the eye axis is no longer parallel to the screen. The resulting expression is:

$$\Delta_{sh}^{XZ} : Fx = \begin{pmatrix} Ex\left(dx\,dz^3 Ix - dx^2 dz^2 Iz - dxdzIxIz^2 + dx^2 Iz^3\right) + Ex^2\left(-dxdz^3 + dx\,dz\,Iz^2\right) \\ + ExEz\left(dx^2 dz^2 - dx^2 Iz^2 - 2|dx|dz^2\tau\right) + Ez\left(2|dx|dz^2 Ix\tau - 2|dx|IxIz^2\tau + 2dxdzIz\tau^2\right) \\ + Ez^2\left(dx|dx|dz\tau + |dx|IxIz\tau - dxdz\tau^2\right) \\ + \left(dx|dx|dz^3\tau - |dx|dz^2 IxIz\tau - dx|dx|dzIz^2\tau + |dx|IxIz^3\tau + dxdz^3\tau^2 - dxdzIz^2\tau^2\right) \end{pmatrix} / w$$

$$Fz = -\begin{pmatrix} ExEz\left(-dxdz\left(Iz^2 - dz^2\right)\right) \\ + Ez\left(Iz^2 - dz^2\right)\left(dxdzIx - dx^2 Iz + 2|dx|Iz\tau\right) \\ + Ez^2\left(Iz^2 - dz^2\right)\left(dx^2 - |dx|\tau\right) \\ + \left(Iz^2 - dz^2\right)\left(|dx|dz^2\tau - |dx|Iz^2\tau\right) \end{pmatrix} / w$$

$$w = Ex\left(-dxdz^3 + dxdzIz^2\right) + Ez\left(dx^2 dz^2 - dx^2 Iz^2 - 2|dx|Iz^2\tau\right) + Ez^2\left(|dx|Iz\tau\right) + \left(dxdz^3 Ix - dx^2 dz^2 Iz - dxdzIxIz^2 + dx^2 Iz^3 - |dx|dz^2 Iz\tau + |dx|Iz^3\tau\right)$$

2.2 Lack of Intersection in General Case

In general the reconstructing rays AG′ and DH′ do not intersect. To solve for the intersection of any two lines in 3-space, one combines several simultaneous equations to yield equations for each coordinate of the intersection point in terms of two coordinates of lines' endpoints. Such equations were used in Appendix B 1.2 and 1.4 where we have Fx in terms of the endpoints' x and z coordinates and Fz in terms of the endpoints' x and z coordinates. These are the intersection coordinates for lines AG′ and DH′ in the construction. These equations find the intersections of the projections of the AG′ and DH′ on the XZ plane. Alternatively for Fz, we could find Fz in terms of the y and z coordinates of the line endpoints. This finds the intersection of the projection of the AG′ and DH′ on the YZ plane. Clearly, if the two 3D lines intersect in 3-space the value for Fz yielded by either solving in terms of YZ or XZ should be the same. If the value differs in general then no 3-space intersection occurs in general. This is in fact what happens. If we solved for Fz either in terms of X or Y, we get an expression of the form:

$$\begin{pmatrix} dQ\,dz^2\,(-dz+Iz)\,(dz+Iz)\,tau - dQ\,Iz^2\,(-dz+Iz)\,(dz+Iz)\,tau + \\ Ez^2\left(dQ\sqrt{dx^2+dy^2}\,(-dz+Iz)\,(dz+Iz) - dQ\,(-dz+Iz)\,(dz+Iz)\,tau\right) + Ez\left(-\sqrt{dx^2+dy^2}\,dz\,EQ\,(-dz+Iz)\,(dz+Iz) + \right. \\ \left. \sqrt{dx^2+dy^2}\,dz\,IQ\,(-dz+Iz)\,(dz+Iz) - dQ\sqrt{dx^2+dy^2}\,Iz\,(-dz+Iz)\,(dz+Iz) + 2\,dQ\,Iz\,(-dz+Iz)\,(dz+Iz)\,tau\right)\Big) / \\ \left(-\sqrt{dx^2+dy^2}\,dz^3\,IQ + dQ\sqrt{dx^2+dy^2}\,dz^2\,Iz + \sqrt{dx^2+dy^2}\,dz\,IQ\,Iz^2 - \right. \\ dQ\sqrt{dx^2+dy^2}\,Iz^3 + EQ\left(\sqrt{dx^2+dy^2}\,dz^3 - \sqrt{dx^2+dy^2}\,dz\,Iz^2\right) + dQ\,dz^2\,Iz\,tau - dQ\,Ez^2\,Iz\,tau - dQ\,Iz^3\,tau + \\ \left. Ez\left(-dQ\sqrt{dx^2+dy^2}\,dz^2 + dQ\sqrt{dx^2+dy^2}\,Iz^2 + 2\,dQ\,Iz^2\,tau\right)\right) \end{pmatrix}$$

where we symbolically substitute Y for Q to get the YZ projected intersection or X for Q to get the XZ projected intersection. Trivially, Ix, dx and Ex are independent from Iy, dy and Ey so the two projected intersection equations are not equal.